SHIFT REGISTERS

Introduction:

A group of flip-flips connected together forms a register. A register is used solely for storing and shifting data which is in the form of 1's and 0's entered from an external source. The Binary information in a register can be moved from stage to stage within the register or into or out of the register upon application of clock pulses. This type of bit movement or shifting essential for certain arithmetic and logic operations used in microprocessors. This gives rise to a group of registers called Shift Registers. They are very important in applications involving the storage and transfer of data in a digital system.

Shift Registers:

Shift Registers consist of a number of single bit "D-Type Data Latches" connected together in a chain arrangement so that the output from one data latch becomes the input of the next latch and so on, thereby moving the stored data serially from either the left or the right direction. The number of individual Data Latches used to make up Shift Registers are determined by the number of bits to be stored with the most common being 8-bits wide. Shift Registers are mainly used to store data and to convert data from either a serial to parallel or parallel to serial format with all the latches being driven by a common clock (Clk) signal making them Synchronous devices. They are generally provided with a Clear or Reset connection so that they can be "SET" or "RESET" as required.

Generally, Shift Registers operate in one of four different modes:

- Serial-in to Parallel-out (SIPO)
- Serial-in to Serial-out (SISO)
- Parallel-in to Parallel-out (PIPO)
- Parallel-in to Serial-out (PISO)
Serial-in to Parallel-out:

4-bit Serial-in to Parallel-out (SIPO) Shift Register:

- Let’s assume that all the flip-flops (FF_A to FF_D) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level "0" i.e., no parallel data output.
- If a logic "1" is connected to the DATA input pin of FF_A then on the first clock pulse the output of FF_A and the resulting Q_A will be set HIGH to logic "1" with all the other outputs remaining LOW at logic "0".
- Assume now that the DATA input pin of FF_A has returned LOW to logic "0". The next clock pulse will change the output of FF_A to logic "0" and the output of FF_B and Q_B HIGH to logic "1".
- The logic "1" has now moved or been "Shifted" one place along the register to the right.
- When the third clock pulse arrives this logic "1" value moves to the output of FF_C (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level "0" because the input has remained at a constant logic level "0".
- The effect of each clock pulse is to shift the DATA contents of each stage one place to the right, and this is shown in the following table until the complete DATA is stored, which can now be read directly from the outputs of Q_A to Q_D. Then the DATA has been converted from a Serial Data signal to a Parallel Data word.

<table>
<thead>
<tr>
<th>Clock Pulse No</th>
<th>Q_A</th>
<th>Q_B</th>
<th>Q_C</th>
<th>Q_D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Serial-in to Serial-out:**

- This Shift Register is very similar to the one above except where as the data was read directly in a parallel form from the outputs Q_A to Q_D, this time the DATA is allowed to flow straight through the register.
- Since there is only one output the DATA leaves the shift register one bit at a time in a serial pattern and hence the name **Serial-in to Serial-Out Shift Register**.

**4-bit Serial-in to Serial-out (SISO) Shift Register**
This type of **Shift Register** also acts as a temporary storage device or as a time delay device, with the amount of time delay being controlled by the number of stages in the register, 4, 8, 16 etc or by varying the application of the clock pulses.

**Parallel-in to Serial-out:**

- Parallel-in to Serial-out Shift Registers act in the opposite way to the Serial-in to Parallel-out one above.
- The DATA is applied in parallel form to the parallel input pins $P_A$ to $P_D$ of the register and is then read out sequentially from the register one bit at a time from $P_A$ to $P_D$ on each clock cycle in a serial format.

**4-bit Parallel-in to Serial-out (PISO) Shift Register**

- As this type of Shift Register converts parallel data, such as an 8-bit data word into serial data it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line.
Parallel-in to Parallel-out:

- Parallel-in to Parallel-out Shift Registers also act as a temporary storage device or as a time delay device. The DATA is presented in a parallel format to the parallel input pins $P_A$ to $P_D$ and then shifts it to the corresponding output pins $Q_A$ to $Q_D$ when the registers are clocked.

4-bit Parallel-in/Parallel-out (PIPO) Shift Register

- As with the Serial-in to Serial-out shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the frequency of the clock pulses.

- "Universal Shift Registers" means it can be used in serial in –serial out, shift left, shift right, serial in –parallel out, parallel in – serial out, and as a parallel-parallel Data Registers.
RING COUNTER (shift register sequencer):

- The ring counter (shift register sequencer) is a unique type of shift register that incorporates feedback from the output of the last flip-flop to the input of the first flip-flop.
- Figure shows a 4-bit ring counter made from D-type flip-flops.
- In this circuit, when the LOAD’ input is set low, Q₀ is forced high by the active-low preset, while Q₁, Q₂, and Q₃ are forced low (cleared) by the active-low clear.
- This causes the binary word 1000 to be stored within the register.
- When the LOAD line is brought low, the data bits stored in the flip-flops are shifted right with each positive clock edge.
- The data bit from the last flip-flop is sent to the D input of the first flip-flop. The shifting cycle will continue to recirculate while the clock is applied.
- To start fresh cycle, the LOAD line is momentarily brought low.

Ring counter using positive edge-triggered D flip-flops
This application can use as lamp decoration or advertiser. You can make more output by those connection use more d-flip flop.

**JOHNSON COUNTER**

- The Johnson shift counter is similar to the ring counter except that its last flip-flop feeds data back to the first flip-flop from its inverted output ($Q$).
- In the simple 4-bit Johnson shift counter shown below, we start out by applying a low to the START line, which sets presets $Q_0$ high and $Q_1$, $Q_2$, and $Q_3$ low $\rightarrow Q_3$ high.
- In other words, you load the register with the binary word 1000, as you did with the ring counter. Now, when you bring START line low, data will shift through the register.
- However, unlike the ring counter, the first bit sent back to the $D0$ input of the first flip-flop will be high because feedback is from $Q3$ not $Q3$.
- At the next clock edge, another high is fed back to $D0$; at the next clock edge, another high is fed back; at the next edge, another high is fed back.
- Only after the fourth clock edge does a low get fed back (the 1 has shifted down to the last flip-flop and $Q3$ goes high).
• At this point, the shift register is full of 1s. As more clock pulses arrive, the feedback loop supplies lows to \( D_0 \) for the next four clock pulses.
• After that, the \( Q \) outputs of all the flip-flops are low while \( Q_3 \) goes high. This high from \( Q_3 \) is fed back to \( D_0 \) during the next positive clock edge, and the cycle repeats.
• As you can see, the 4-bit Johnson shift counter has 8 output stages (which require 8 clock pulses to recycle), not 4, as was the case with the ring counter. Johnson counter using positive edge-triggered D flip-flops

This application can use as LAMP decoration or Advertiser. You can make more output by those connection use mode D- Flip flop.